Coloring Register Pairs (1992) (Make Corrections) (3 citations)
Preston Briggs, Keith D. Cooper, Linda Torczon
ACM Letters on Programming Languages and Systems

CiteSeer Home/Search Bookmark Context Related

Links: ACM DBLP

View or download:
rice.edu/MSCP/pape...loplas.pairs.ps.gz
Cached: PS.gz PS PDF
Image Update Help

From: rice.edu/MSCP/publications (more)
(Enter author homepages)

(Enter summary)

Rate this article: 1 2 3 4 5 (best)
Comment on this article

Abstract: This paper discusses the representation of register pairs in a graph coloring allocator. It explains the problems that arise with Chaitin's allocator and shows how the optimistic allocator avoids them. It provides a rationale for determining how to add larger aggregates to the interference graph. (Update)

Context of citations to this paper: More

...on graph coloring register allocation is not entirely devoid of work on irregular architectures. **In particular, Nickerson [9] and Briggs [2,3] both describe approaches that adapt graph coloring based allocators for regular architectures to a limited class of irregular**...

Cited by: More
A Progressive Register Allocator for Irregular Architectures - David Koes And (Correct)
Graph-Coloring Register Allocation for Irregular Architectures - Smith, Holloway (2001) (Correct)

Similar documents (at the sentence level):
52.1%: Register Allocation via Graph Coloring - Briggs (1992) (Correct)

Active bibliography (related documents): More All
0.4: Using Compiler Technology to Drive Advanced Microprocessors - Cooper (1992) (Correct)
0.2: Improvements to Graph Coloring Register Allocation - Briggs, Cooper, Torczon (1994) (Correct)
0.2: Instruction-Processing Optimization Techniques For VLSI.. - Bunda (1993) (Correct)

Similar documents based on text: More All
0.9: Live Range Splitting in a Graph Coloring Register Allocator - Cooper, Simpson (1998) (Correct)
0.6: Reducing the Impact of Spill Code - Harvey (Correct)
0.5: Rematerialization - Briggs, Cooper, Torczon (1992) (Correct)

Related documents from co-citation: More All
3: Register Allocation via Graph Coloring - Briggs - 1992
3: Precise Register Allocation for Irregular Architectures - Kong, Wilken - 1998
2: Register allocation for irregular architectures (context) - Scholz, Eckstein - 2002

BibTeX entry: (Update)

P. Briggs, K. Cooper, and L. Torczon, "Coloring Register Pairs," ACM Letters on Programming Languages and Systems, 1(1):3--13, March 1992. http://citeseer.ist.psu.edu/briggs92coloring.html More

```
@article{ briggs92coloring,
    author = "Preston Briggs and Keith D. Cooper and Linda Torczon",
    title = "Coloring Register Pairs",
    journal = "ACM Letters on Programming Languages and Systems",
    volume = "1",
    number = "1",
    month = "March",
    publisher = "ACM Press",
    pages = "3--13",
```

Sign in

Google Web Images Video News Maps more »
Spills reloading instructions parallel architectu Search Advanced Search Preferences

Web Results **1 - 10** of about **31,600** for **Spills reloading instructions parallel architecture**. (**0.13** seconds)

[PDF] A **Spill** Code Minimization Technique—Application in the Metrowerks ...
File Format: PDF/Adobe Acrobat
operations can be executed in **parallel**. Examples of such **architectures** are ... Change the rematerialization **instruction** to a **reload instruction**; ...
www.springerlink.com/index/P325367322070258.pdf - Similar pages

[PDF] The Named-State Register File: implementation and performanc ...
File Format: PDF/Adobe Acrobat
Parallel. U. FIGURE 13. Register **spill** and **reload** overhead as a ... approach to adding registers into **instruction** set. **architectures**. ...
ieeexplore.ieee.org/iel2/3040/8763/00386560.pdf?arnumber=386560 - Similar pages

[PDF] S P C ' M A
File Format: PDF/Adobe Acrobat
word **spill** area to **spill** and **reload** the entire. register. ... scheduling **architecture**. The data-**parallel** select **architecture** inte- ...
ieeexplore.ieee.org/iel5/40/34097/01624323.pdf - Similar pages

[PDF] A Design Space Exploration Framework for Reduced Bit-width ...
File Format: PDF/Adobe Acrobat
The presence of extra **instructions** (due to extend, **spill/reload**,. etc.) in rISAcode negatively impacts performance. For. the rISA hybrid **architecture**, ...
portal.acm.org/ft_gateway.cfm?id=581228&
type=pdf&coll=&dl=acm&CFID=15151515&CFTOKEN=6... - Similar pages

[PPT] Computer **Architecture** and Organization
File Format: Microsoft Powerpoint - View as HTML
Sequential simulation. **Parallel** simulation. **Architecture** description ... **Spill/reload** code inserted after scheduling which requires an extra scheduling step ...
www.elis.rug.ac.be/ntca/Presentations/corporaal.ppt - Similar pages

Synergistic Processing in Cell's Multicore **Architecture**
Data-**parallel instructions** support data-level parallelism, ... register allocation to a full quadword **spill** area to **spill** and **reload** the entire register. ...
doi.ieeecomputersociety.org/10.1109/MM.2006.41 - Similar pages

[PDF] Optimal Spilling for CISC Machines with Few Registers
File Format: PDF/Adobe Acrobat - View as HTML
as we cannot insert **spill** or **reload instructions** at P2. We therefore ... **Parallel Architecture** and Compilation Techniques,. pages. 196-204, 1998. ...
www.cs.wisc.edu/~fischer/cs701.f06/pldi01-cisc.spilling.pdf - Similar pages

[PDF] EPIC-1 Workshop on Explicitly **Parallel Instruction** Computing (EPIC ...
File Format: PDF/Adobe Acrobat
architecture's **parallel** compare **instructions**. Using AND-type ... In addition, **reloading**. from the register that was chosen as **spill** location could avoid ...
systems.cs.colorado.edu/EPIC1/EPIC1.pdf - Similar pages

[PDF] Demystifying On-the-Fly **Spill** Code
File Format: PDF/Adobe Acrobat - View as HTML

PORTAL
USPTO
Subscribe (Full Service) Register (Limited Service, Free) Login
Search: The ACM Digital Library The Guide
parallel register <and> spill <and> reload instructions
SEARCH
THE ACM DIGITAL LIBRARY
Feedback Report a problem Satisfaction survey

Terms used **parallel register** and **spill** and **reload instructions** Found **85,744** of **196,064**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Optimal spilling for CISC machines with few registers

Andrew W. Appel, Lal George
May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation PLDI '01**, Volume 36 Issue 5
Publisher: ACM Press

Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, citings, index terms

Many graph-coloring register-allocation algorithms don't work well for machines with few registers. Heuristics for live-range splitting are complex or suboptimal; heuristics for register assignment rarely factor the presence of fancy addressing modes; these problems are more severe the fewer registers there are to work with. We show how to optimally split live ranges and optimally use addressing modes, where the optimality condition measures dynamically weighted loads and stores but not regis ...

2 Optimizing compiler for shared-memory multiple SIMD architecture

Weihua Zhang, Xinglong Qian, Ye Wang, Binyu Zang, Chuanqi Zhu
June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN/SIGBED conference on Language, compilers and tool support for embedded systems LCTES '06**, Volume 41 Issue 7
Publisher: ACM Press

Full text available: pdf(375.75 KB) Additional Information: full citation, abstract, references, index terms

With the rapid growth of multimedia and game, these applications put more and more pressure on the processing ability of modern processors. Multiple SIMD architecture is widely used in multimedia processing field as a multimedia accelerator.With the consideration of power consumption and chip size, shared memory multiple SIMD architecture is mainly used in embedded SOCs. In order to further fit mobile environment, there is the constraint of limited register number as well. Although shared memory ...

Keywords: locality, multiple SIMD, optimization, replication, shared memory

3 Compilers: A framework for reducing instruction scheduling overhead in dynamic compilers

Vikki Tang, Joran Siu, Alexander Vasilevskiy, Marcel Mitran
October 2006 **Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research CASCON '06**

Publisher: ACM Press
Full text available: pdf(139.39 KB) htm(1.81 KB) Additional Information: full citation, abstract, references

Start-up time is a serious concern for high-availability applications such as web servers, transaction managers, and batch processes. Compilation time contributes directly to start-up costs in dynamic compilers. Up to 30% of compilation time is spent scheduling instructions in the IBM® Testarossa just-in-time compiler. In this paper, we describe a scheduling framework that reduces scheduling overhead by up to 61% with little to no degradation in throughput performance. By combining online pr ...

4 A global progressive register allocator
David Ryan Koes, Seth Copen Goldstein
June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06**, Volume 41 Issue 6
Publisher: ACM Press
Full text available: pdf(721.32 KB) Additional Information: full citation, abstract, references, index terms

This paper describes a global progressive register allocator, a register allocator that uses an expressive model of the register allocation problem to quickly find a good allocation and then progressively find better allocations until a provably optimal solution is found or a preset time limit is reached. The key contributions of this paper are an expressive model of global register allocation based on multicommodity network flows that explicitly represents spill code optimization, register pref ...

Keywords: progressive solver, register alocation

5 An experimental study of several cooperative register allocation and instruction scheduling strategies
Cindy Norris, Lori L. Pollock
December 1995 **Proceedings of the 28th annual international symposium on Microarchitecture MICRO 28**
Publisher: IEEE Computer Society Press
Full text available: pdf(1.17 MB) Additional Information: full citation, references, citings, index terms

6 Instruction selection, resource allocation, and scheduling in the AVIV retargetable code generator
Silvina Hanono, Srinivas Devadas
May 1998 **Proceedings of the 35th annual conference on Design automation DAC '98**
Publisher: ACM Press
Full text available: pdf(214.25 KB) Additional Information: full citation, abstract, references, citings, index terms

The AVIV retargetable code generator produces optimized machine code for target processors with different instruction set architectures AVIV optimizes for minimum code size. Retargetable code generation requires the development of heuristic algorithms for instruction selection, resource allocation, and scheduling. AVIV addresses these code generation subproblems concurrently, whereas most current code generation systems address them sequentially. It accomplishes this by convertin ...

Keywords: MPEG4, codec, design automatian, flip-flops, level converters, low power, placement, synthesis, voltage scaling

7 Formal certification of a compiler back-end or: programming a compiler with a proof assistant

Xavier Leroy

January 2006 **ACM SIGPLAN Notices , Conference record of the 33rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '06**, Volume 41 Issue 1

Publisher: ACM Press

Full text available: pdf(187.24 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper reports on the development and formal certification (proof of semantic preservation) of a compiler from Cminor (a C-like imperative language) to PowerPC assembly code, using the Coq proof assistant both for programming the compiler and for proving its correctness. Such a certified compiler is useful in the context of formal methods applied to the certification of critical software: the certification of the compiler guarantees that the safety properties proved on the source code hold f ...

Keywords: certified compilation, compiler transformations and optimizations, program proof, semantic preservation, the Coq theorem prover

8 Improvements to graph coloring register allocation

Preston Briggs, Keith D. Cooper, Linda Torczon

May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 3

Publisher: ACM Press

Full text available: pdf(2.00 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We describe two improvements to Chaitin-style graph coloring register allocators. The first, optimistic coloring, uses a stronger heuristic to find a k-coloring for the interference graph. The second extends Chaitin's treatment of rematerialization to handle a larger class of values. These techniques are complementary. Optimistic coloring decreases the number of procedures that require spill code and reduces the amount of spill code when sp ...

Keywords: code generation, graph coloring, register allocation

9 A comparison of software and hardware techniques for x86 virtualization

Keith Adams, Ole Agesen

October 2006 **ACM SIGOPS Operating Systems Review , ACM SIGARCH Computer Architecture News , ACM SIGPLAN Notices , Proceedings of the 12th international conference on Architectural support for programming languages and operating systems ASPLOS-XII**, Volume 40 , 34 , 41 Issue 5 , 5 , 11

Publisher: ACM Press

Full text available: pdf(236.66 KB) Additional Information: full citation, abstract, references, index terms

Until recently, the x86 architecture has not permitted classical trap-and-emulate virtualization. Virtual Machine Monitors for x86, such as VMware ® Workstation and Virtual PC, have instead used binary translation of the guest kernel code. However, both Intel and AMD have now introduced architectural extensions to support classical virtualization.We compare an existing software VMM with a new VMM designed for the emerging hardware support. Surprisingly, the hardware VMM often suffers lower ...

Keywords: MMU, SVM, TLB, VT, dynamic binary translation, nested paging, virtual machine monitor, virtualization, x86

10 Demystifying on-the-fly spill code

Alex Aletà, Josep M. Codina, Antonio Gonzàlez, David Kaeli

June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation PLDI '05**, Volume 40 Issue 6

Publisher: ACM Press

Full text available: pdf(192.08 KB) Additional Information: full citation, abstract, references, index terms

Modulo scheduling is an effective code generation technique that exploits the parallelism in program loops by overlapping iterations. One drawback of this optimization is that register requirements increase significantly because values across different loop iterations can be live concurrently. One possible solution to reduce register pressure is to insert spill code to release registers. Spill code stores values to memory between the producer and consumer instructions.Spilling heuristics can be ...

Keywords: modulo scheduling, register allocation, spill code

11 A schedular-sensitive global register allocator

C. Norris, L. L. Pollock

December 1993 **Proceedings of the 1993 ACM/IEEE conference on Supercomputing Supercomputing '93**

Publisher: ACM Press

Full text available: pdf(1.08 MB) Additional Information: full citation, references, citings, index terms

12 Reducing Instruction Fetch Cost by Packing Instructions into RegisterWindows

Stephen Hines, Gary Tyson, David Whalley

November 2005 **Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture MICRO 38**

Publisher: IEEE Computer Society

Full text available: pdf(706.27 KB) Publisher Site Additional Information: full citation, abstract, index terms

Instruction packing is a combination compiler/ architectural approach that allows for decreased code size, reduced power consumption and improved performance. The packing is obtained by placing frequently occurring instructions into an Instruction Register File (IRF). Multiple IRF entries can then be accessed using special packed instructions. Previous IRF efforts focused on using a single 32-entry register file for the duration of an application. This paper presents software and hardware extens ...

13 Dependences and register allocation: Prematerialization: reducing register pressure for free

Ivan D. Baev, Richard E. Hank, David H. Gross

September 2006 **Proceedings of the 15th international conference on Parallel architectures and compilation techniques PACT '06**

Publisher: ACM Press

Full text available: pdf(159.54 KB) Additional Information: full citation, abstract, references, index terms

Modern compiler transformations that eliminate redundant computations or reorder instructions, such as partial redundancy elimination and instruction scheduling, are very effective in improving application performance but tend to create longer and potentially more complex live ranges. Typically the task of dealing with the increased register pressure is left to the register allocator. To avoid introduction of spill code which can

reduce or completely eliminate the benefit of earlier optimization ...

Keywords: Itanium, VLIW, register allocation, register pressure, rematerialization

14 Dual-issue scheduling with spills for binary trees

Waleed M. Meleis, Edward S. Davidson

January 1999 **Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms SODA '99**

Publisher: Society for Industrial and Applied Mathematics

Full text available: pdf(972.47 KB) Additional Information: full citation, references, index terms

15 Interprocedural register allocation for lazy functional languages

Urban Boquist

October 1995 **Proceedings of the seventh international conference on Functional programming languages and computer architecture FPCA '95**

Publisher: ACM Press

Full text available: pdf(1.49 MB) Additional Information: full citation, references, citings, index terms

16 Compilation framework for code size reduction using reduced bit-width ISAs (rISAs)

Aviral Shrivastava, Partha Biswas, Ashok Halambi, Nikil Dutt, Alex Nicolau

January 2006 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 11 Issue 1

Publisher: ACM Press

Full text available: pdf(982.75 KB) Additional Information: full citation, abstract, references, citings, index terms

For many embedded applications, program code size is a critical design factor. One promising approach for reducing code size is to employ a "dual instruction set", where processor architectures support a normal (usually 32-bit) Instruction Set, and a narrow, space-efficient (usually 16-bit) Instruction Set with a limited set of opcodes and access to a limited set of registers. This feature however, requires compilers that can reduce code size by compiling for both Instruction Sets. E ...

Keywords: Code generation, code compression, codesize reduction, compilers, dual instruction set, narrow bit-width instruction set, optimization, rISA, register pressure-based code generation, retargetable compilers, thumb

17 Code generator optimizations for the ST120 DSP-MCU core

B. Dupont de Dinechin, F. de Ferri, C. Guillon, A. Stoutchinin

November 2000 **Proceedings of the 2000 international conference on Compilers, architecture, and synthesis for embedded systems CASES '00**

Publisher: ACM Press

Full text available: pdf(426.51 KB) Additional Information: full citation, citings

18 An Efficient Compiler Technique for Code Size Reduction Using Reduced Bit-Width ISAs

A. Halambi, A. Shrivastava, P. Biswas, N. Dutt, A. Nicolau

March 2002 **Proceedings of the conference on Design, automation and test in Europe DATE '02**

Publisher: IEEE Computer Society
Full text available: pdf(289.61 KB) Additional Information: full citation, abstract, citings

For many embedded applications, program code sizeis a critical design factor. One promising approach forreducing code size is to employ a "dual instruction set",where processor architectures support a normal (usually32 bit) Instruction Set, and a narrow, space-efficient(usually 16 bit) Instruction Set with a limited set of op-codes and access to a limited set of registers. This feature, however, requires compilers that can reduce codesize by compiling for both Instruction Sets. Existingcompiler ...

19 Reorganizing global schedules for register allocation

Gang Chen, Michael D. Smith
May 1999 **Proceedings of the 13th international conference on Supercomputing ICS '99**
Publisher: ACM Press
Full text available: pdf(1.15 MB) Additional Information: full citation, references, citings, index terms

Keywords: instruction-level parallelism, register allocation, superblock scheduling

20 Unified management of registers and cache using liveness and cache bypass

C.-H. Chi, H. Dietz
June 1989 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1989 Conference on Programming language design and implementation PLDI '89**, Volume 24 Issue 7
Publisher: ACM Press
Full text available: pdf(1.20 MB) Additional Information: full citation, abstract, references, citings, index terms

In current computer memory system hierarchy, registers and cache are both used to bridge the reference delay gap between the fast processor(s) and the slow main memory. While registers are managed by the compiler using program flow analysis, cache is mainly controlled by hardware without any program understanding. Due to the lack of coordination in managing these two memory structures, significant loss of system performance results because: Cache space is wasted to hold in ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

register allocator <and> spill <and> reload instructions SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **register allocator** and **spill** and **reload instructions** Found **57,139** of **196,064**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 A global progressive register allocator

David Ryan Koes, Seth Copen Goldstein

June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06**, Volume 41 Issue 6

Publisher: ACM Press

Full text available: pdf(721.32 KB) Additional Information: full citation, abstract, references, index terms

This paper describes a global progressive register allocator, a register allocator that uses an expressive model of the register allocation problem to quickly find a good allocation and then progressively find better allocations until a provably optimal solution is found or a preset time limit is reached. The key contributions of this paper are an expressive model of global register allocation based on multicommodity network flows that explicitly represents spill code optimization, register pref ...

Keywords: progressive solver, register alocation

2 Dynamic compilation techniques: Optimized interval splitting in a linear scan register allocator

Christian Wimmer, Hanspeter Mössenböck

June 2005 **Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments VEE '05**

Publisher: ACM Press

Full text available: pdf(341.54 KB) Additional Information: full citation, abstract, references, citings, index terms

We present an optimized implementation of the linear scan register allocation algorithm for Sun Microsystems' Java HotSpot™ client compiler. Linear scan register allocation is especially suitable for just-in-time compilers because it is faster than the common graph-coloring approach and yields results of nearly the same quality.Our allocator improves the basic linear scan algorithm by adding more advanced optimizations: It makes use of lifetime holes, splits intervals if the register press ...

Keywords: compilers, graph-coloring, java, just-in-time compilation, linear scan, optimization, register allocation

3 Optimal spilling for CISC machines with few registers

Andrew W. Appel, Lal George
May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation PLDI '01**, Volume 36 Issue 5
Publisher: ACM Press
Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, citings, index terms

Many graph-coloring register-allocation algorithms don't work well for machines with few registers. Heuristics for live-range splitting are complex or suboptimal; heuristics for register assignment rarely factor the presence of fancy addressing modes; these problems are more severe the fewer registers there are to work with. We show how to optimally split live ranges and optimally use addressing modes, where the optimality condition measures dynamically weighted loads and stores but not regis ...

4 A schedular-sensitive global register allocator
C. Norris, L. L. Pollock
December 1993 **Proceedings of the 1993 ACM/IEEE conference on Supercomputing Supercomputing '93**
Publisher: ACM Press
Full text available: pdf(1.08 MB) Additional Information: full citation, references, citings, index terms

5 Integrating register allocation and instruction scheduling for RISCs
David G. Bradlee, Susan J. Eggers, Robert R. Henry
April 1991 **ACM SIGARCH Computer Architecture News , ACM SIGOPS Operating Systems Review , ACM SIGPLAN Notices , Proceedings of the fourth international conference on Architectural support for programming languages and operating systems ASPLOS-IV**, Volume 19 , 25 , 26 Issue 2 , Special Issue , 4
Publisher: ACM Press
Full text available: pdf(1.11 MB) Additional Information: full citation, references, citings, index terms

6 Improvements to graph coloring register allocation
Preston Briggs, Keith D. Cooper, Linda Torczon
May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 3
Publisher: ACM Press
Full text available: pdf(2.00 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We describe two improvements to Chaitin-style graph coloring register allocators. The first, optimistic coloring, uses a stronger heuristic to find a k-coloring for the interference graph. The second extends Chaitin's treatment of rematerialization to handle a larger class of values. These techniques are complementary. Optimistic coloring decreases the number of procedures that require spill code and reduces the amount of spill code when sp ...

Keywords: code generation, graph coloring, register allocation

7 Register windows vs. register allocation
D. W. Wall
June 1988 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1988 conference on Programming Language design and Implementation PLDI '88**, Volume 23 Issue 7

Publisher: ACM Press

Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, citings, index terms

A large register set can be exploited by keeping variables and constants in registers instead of in memory. Hardware register windows and compile-time or link-time global register allocation are ways to do this. A measure of the effectiveness of any of these register management schemes is how thoroughly they remove loads and stores. This measure must also count extra loads and stores executed because of window overflow or conflicts between procedures. By combining profiling, inst ...

8 An experimental study of several cooperative register allocation and instruction scheduling strategies

Cindy Norris, Lori L. Pollock

December 1995 **Proceedings of the 28th annual international symposium on Microarchitecture MICRO 28**

Publisher: IEEE Computer Society Press

Full text available: pdf(1.17 MB) Additional Information: full citation, references, citings, index terms

9 Spill code minimization via interference region spilling

Peter Bergner, Peter Dahl, David Engebretsen, Matthew O'Keefe

May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation PLDI '97**, Volume 32 Issue 5

Publisher: ACM Press

Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings, index terms

Many optimizing compilers perform global register allocation using a Chaitin-style graph coloring algorithm. Live ranges that cannot be allocated to registers are spilled to memory. The amount of code required to spill the live range depends on the spilling heuristic used. Chaitin's spilling heuristic offers some guidance in reducing the amount of spill code produced. However, this heuristic does not allow the partial spilling of live ranges and the reduction in spill code is limited to a local ...

10 Quality and speed in linear-scan register allocation

Omri Traub, Glenn Holloway, Michael D. Smith

May 1998 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation PLDI '98**, Volume 33 Issue 5

Publisher: ACM Press

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

A *linear-scan* algorithm directs the global allocation of register candidates to registers based on a simple linear sweep over the program being compiled. This approach to register allocation makes sense for systems, such as those for dynamic compilation, where compilation speed is important. In contrast, most commercial and research optimizing compilers rely on a graph-coloring approach to global register allocation. In this paper, we compare the performance of a linear-scan method agains ...

Keywords: binpacking, global register allocation, graph coloring, linear scan

11 Interprocedural register allocation for lazy functional languages

Urban Boquist

October 1995 **Proceedings of the seventh international conference on Functional programming languages and computer architecture FPCA '95**
Publisher: ACM Press
Full text available: pdf(1.49 MB) Additional Information: full citation, references, citings, index terms

12 The priority-based coloring approach to register allocation
Fred C. Chow, John L. Hennessy
October 1990 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 12 Issue 4
Publisher: ACM Press
Full text available: pdf(2.97 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Global register allocation plays a major role in determining the efficacy of an optimizing compiler. Graph coloring has been used as the central paradigm for register allocation in modern compilers. A straightforward coloring approach can suffer from several shortcomings. These shortcomings are addressed in this paper by coloring the graph using a priority ordering. A natural method for dealing with the spilling emerges from this approach. The detailed algorithms for a priority-based colori ...

13 Compilation and run-time systems: A faster optimal register allocator
Changqing Fu, Kent Wilken
November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture MICRO 35**
Publisher: IEEE Computer Society Press
Full text available: pdf(982.37 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Recently researchers have proposed modeling register allocation as an integer linear programming (IP) problem and solving it optimally for general purpose processors [17, 20] and for dedicated embedded systems [23]. Compared with traditional graph-coloring approaches, the IP-based allocators can improve a program's performance. However, the solution times are much slower.This paper presents an IP-based optimal register allocator which is much faster than previous work. We present several local a ...

14 Dependences and register allocation: Prematerialization: reducing register pressure for free
Ivan D. Baev, Richard E. Hank, David H. Gross
September 2006 **Proceedings of the 15th international conference on Parallel architectures and compilation techniques PACT '06**
Publisher: ACM Press
Full text available: pdf(159.54 KB) Additional Information: full citation, abstract, references, index terms

Modern compiler transformations that eliminate redundant computations or reorder instructions, such as partial redundancy elimination and instruction scheduling, are very effective in improving application performance but tend to create longer and potentially more complex live ranges. Typically the task of dealing with the increased register pressure is left to the register allocator. To avoid introduction of spill code which can reduce or completely eliminate the benefit of earlier optimization ...

Keywords: Itanium, VLIW, register allocation, register pressure, rematerialization

15 Demand-driven register allocation
Todd A. Proebsting, Charles N. Fischer

November 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 6
Publisher: ACM Press

Full text available: pdf(305.88 KB) Additional Information: full citation, abstract, references, citings, index terms, review

A new global register allocation technique, demand-driven register allocation, is described. Demand-driven register allocation quantifies the costs and benefits of allocating variables to registers over live ranges so that high-quality allocations can be made. Local allocation is done first, and then global allocation is done iteratively beginning in the most deeply nested loops. Because local allocation precedes global allocation, demand-driven allocation does not interfer ...

Keywords: optimizing compiler

16 Experience with a software-defined machine architecture

David W. Wall
May 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 14 Issue 3
Publisher: ACM Press

Full text available: pdf(2.86 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We have built a system in which the compiler back end and the linker work together to present an abstract machine at a considerably higher level than the actual machine. The intermediate language translated by the back end is the target language of all high-level compilers and is also the only assembly language generally available. This lets us do intermodule register allocation, which would be harder if some of the code in the program had come from a traditional assembler, out of sight of ...

Keywords: RISC, graph coloring, intermediate language, interprocedural, optimization, pipeline scheduling, profiling, register allocation, register windows

17 Iterated register coalescing

Lal George, Andrew W. Appel
May 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 3
Publisher: ACM Press

Full text available: pdf(273.50 KB) Additional Information: full citation, abstract, references, citings, index terms

An important function of any register allocator is to target registers so as to eliminate copy instructions. Graph-coloring register allocation is an elegant approach to this problem. If the source and destination of a move instruction do not interfere, then their nodes can be coalesced in the interference graph. Chaitin's coalescing heuristic could make a graph uncolorable (i.e., introduce spills); Briggs et al. demonstrated a conservative coalescing heuristic that preserves colorability. ...

Keywords: copy propagation, graph coloring, register allocation, register coalescing

18 Rematerialization

Preston Briggs, Keith D. Cooper, Linda Torczon
July 1992 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1992 conference on Programming language design and implementation PLDI '92**, Volume 27 Issue 7

Publisher: ACM Press

Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper examines a problem that arises during global register allocation – rematerialization. If a value cannot be kept in a register, the allocator should recognize when it is cheaper to recompute the value (rematerialize it) than to store and reload it. Chaitin's original graph-coloring allocator handled simple instance of this problem correctly. This paper details a general solution to the problem and presents experimental evidence that shows its importance.

19 Graph coloring register allocation for processors with multi-register operands

Brian R. Nickerson

June 1990 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1990 conference on Programming language design and implementation PLDI '90**, Volume 25 Issue 6

Publisher: ACM Press

Full text available: pdf(1.41 MB) Additional Information: full citation, abstract, references, citings, index terms

Though graph coloring algorithms have been shown to work well when applied to register allocation problems, the technique has not been generalized for processor architectures in which some instructions refer to individual operands that are comprised of multiple registers. This paper presents a suitable generalization.

20 Iterated register coalescing

Lal George, Andrew W. Appel

January 1996 **Proceedings of the 23rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '96**

Publisher: ACM Press

Full text available: pdf(955.09 KB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player